МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Национальный исследовательский**

**Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**ЛАБОРАТОРНАЯ РАБОТА №7**

**«Структура хранения данных: односвязный линейный список с использованием указателей»**

**Выполнил:** студент группы 381908-4

Яшин Егор Олегович

**Проверил:**

ассистент каф. МОСТ ИИТММ

Лебедев Илья Геннадьевич

Нижний Новгород

2020.

Оглавление

[1. Введение 3](#_Toc2009015)

[2. Постановка задачи 4](#_Toc2009016)

[3. Руководство пользователя 5](#_Toc2009017)

[4. Руководство программиста 6](#_Toc2009018)

[5.1. Описание структуры программы 6](#_Toc2009019)

[5.2. Описание структур данных 6](#_Toc2009020)

[5.3. Описание алгоритмов 7](#_Toc2009021)

[5. Заключение 9](#_Toc2009022)

[6. Список литературы 10](#_Toc2009023)

# Введение

Односвязный список – это структура данных, представляющая собой упорядоченный набор значений, в котором некоторое значение может встречаться более одного раза.

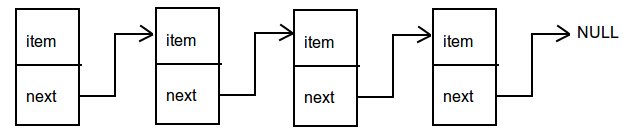
Начало списка называют головным элементом, а звенья списка - узлами. Каждый узел односвязного списка помимо лежащего в нем значения, содержит поле указателя на следующий узел. Поле указателя последнего узла содержит нулевое значение (указывает на NULL).

Рисунок 1. Структура списка

В односвязном списке вставка и удаление узлов производится либо в начало, либо в конец списка.

Структура списка ограничивает доступ к его узлам по индексу. Список нельзя индексировать, как массив. Чтобы попасть на некоторый узел односвязного списка, необходимо последовательно пройти весь путь от головного элемента до нужного узла.

Цель данной лабораторной работы – реализация структуры данных односвязного списка.

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

1. Разработка и реализация класса списка - TList
2. Реализация тестов на базе Google Test
3. Пример использования класса TList

# Руководство пользователя

Пользователю предлагается рассмотреть реализацию списка, заполнение данными происходит автоматически, передаются значения 10,20,30 , а после выводится список



# Руководство программиста

## Описание структуры программы

В программе содержатся следующие модули:

* Модуль List – содержит заголовочные файлы List.h, в котором описаны и реализованы классы TList и TListElem, и файлы List.cpp
* Модуль List – содержит файл main\_List.cpp, в котором реализован пример использования класса TList;
* Модуль MP2\_TESTS– содержит файл test\_List.cpp, в котором реализованы тесты для классов TElem и Tlist

## Описание структур данных

* Класс TElem

Класс TListElem является шаблонным.

T data; - переменная, хранящая элемент списка

TListElem\* next=0; - указатель на следующий элемент списка

TListElem\* prev=0;- указатель на предыдущий элемент списка

TListElem(T \_data=0);- конструктор.

~TListElem();- деструктор.

TListElem<T>\* GetNext();- метод, возвращающий указатель на следующий элемент.

TListElem<T>\* GetPrev();- метод, возвращающий указатель на предыдущий элемент.

T GetData() - метод, возвращающий значение элемента.

void SetData(T \_data) - метод, позволяющий установить значение элемента.

void SetNext(TListElem\* \_next); – метод, позволяющий установить указатель на следующий элемент списка.

void SetPrev(TListElem\* \_prev); – метод, позволяющий установить указатель на предыдущий элемент списка.

операторы ввода и вывода

template <class T1>

friend ostream& operator<< (ostream& ostr, const TListElem<T1>& A);

template <class T1>

friend istream& operator >> (istream& istr, TListElem<T1>& A);

* Класс TList

TListElem<T>\* root=0; – указатель на начало списка.

int count;– число элементов в списке.

TList() - конструктор .

TList(TList<T>& \_v);- конструктор копирования.

~TList(); - деструктор.

TList<T>& operator =(TList<T>& \_v); -оператор присваивания

bool IsEmpty(void) const; -проверка на пустоту

bool IsFull(void) const; -проверка на полноту

void InsFirst(T d);

void InsLast(T d);

void Ins(TListElem<T>\* e, T d);

TListElem<T>\* GetFirst();

TListElem<T>\* GetLast();

void DelFirst();

void DelLast();

void Del(TListElem<T>\* e);

загрузка из файла и операции вывода и ввода

void LoadToFile();

template <class T1>

friend ostream& operator<< (ostream& ostr, const TList<T1>& A);

template <class T1>

friend istream& operator >> (istream& istr, TList<T1>& A);

## Описание алгоритмов

**Добавление элемента в начало списка.**

При добавлении элемента в начало списка мы создаем указатель на объект класса TElem, далее выделяем память под объект этого класса и создаем элемент списка. Указатель на начало списка переопределяем на созданный элемент списка. Алгоритм работает за линейное время, сложность O(n).

**Удаление элемента из начала списка.**

При удалении элемента из начала списка (если он не пуст) создаем указательна объект класса TElem, которому присваиваем значение текущего начала списка. Создаем временную переменную*,* в которую записываем значение первого элемента. Начало списка устанавливаем на следующий за удаляемым элемент. Удаляем указатель, указывающий на старое начало списка. Алгоритм работает за линейное время, сложность O(n).

**Добавление элемента в конец списка.**

Проверяем есть ли элементы в списке. Если есть, то создаем указатель на объект класса TElem, в него записываем значение начала списка. В цикле ищем последний элемент*.* Затем выделяем память под новый элемент списка и создаем его. Устанавливаем для найденного последнего элемента указатель на созданный элемент. Если в списке нет элементов, то создаем элемент и указателю на начало списка присваиваем значение, указывающее на созданный элемент.

# Заключение

В ходе выполнения лабораторной был произведен анализ задачи - установлено понятие списка на указателях. Была разработана библиотека, реализующая шаблонный класс списка. Она позволяет при работе со списком выполнять базовые операции извлечения/добавление элементов списка.

Были реализованы тесты для проверки работоспособности класса стека на базе GoogleTest.

# Список литературы

1. Википедия. Статья «Список»: [https://ru.wikipedia.org/wiki/Список\_(информатика)].
2. Васильев А.Н. Самоучитель С++ с примерами и задачами. -СПб.: Наука и Техника, 2016. -480с.
3. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.